

UNIT, METHOD AND COMPUTER PROGRAM PRODUCT FOR NUMERICAL OPERATION

Publication number: JP9231054

Publication date: 1997-09-05

Inventor: MORI HIROSHI; SAITO KAZUHIRO; MAKITA TAKESHI;
YAMADA OSAMU

Applicant: CANON KK

Classification:

- International: G06F7/38; G06T1/00; H04N1/403; G06F7/38;
G06T1/00; H04N1/403; (IPC1-7): G06F7/38; G06T1/00;
H04N1/403

- european:

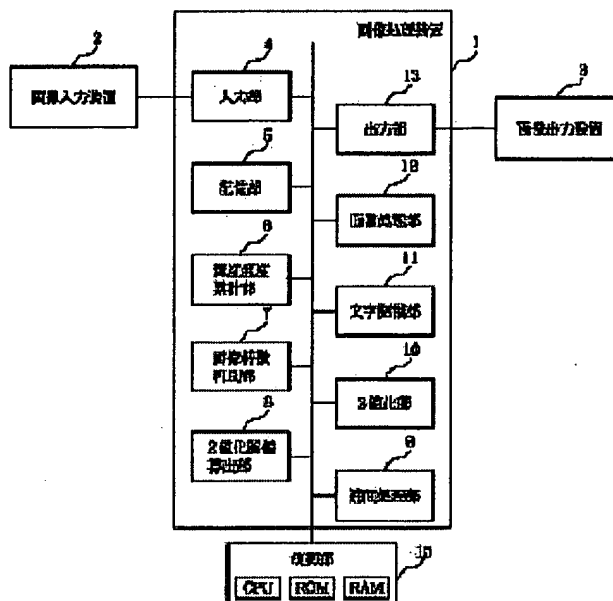
Application number: JP19960041638 19960228

Priority number(s): JP19960041638 19960228

Report a data error here

Abstract of JP9231054

PROBLEM TO BE SOLVED: To maintain fixed accuracy within a valid numerical value handleable by a computer and to accelerate a repetitive arithmetic operation especially by rounding down the digit number of an arithmetic value when it is predicted that the arithmetic value is to be larger than a set valid digit number and carrying the digit number of the arithmetic value when it is predicted that it is to be smaller. **SOLUTION:** Character blocks are discriminated for inputted multilevel images, respectively optimum binarization threshold values are decided and binary images are generated by the binarization threshold value. In the numerical operation processing of a control part 15, the accuracy of a part under a decimal point is maintained even in an integer operation by multiplying a valid digit number preservation variable decided by the processing environment of the computer to the arithmetic value and an arithmetic operation is performed without making an arithmetic result exceed an integer valid numerical value handleable by the computer by dividing the arithmetic value by the valid digit number preservation variable obtained for respective units with average value calculation and skew value calculation as one unit. Thus, the fixed accuracy is maintained and the numerical operation method of an image processing at a high speed is obtained.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-231054

(43) 公開日 平成9年(1997)9月5日

(51) Int.Cl. ^a	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F	7/38		G 0 6 F 7/38	B
G 0 6 T	1/00		15/62	A
H 0 4 N	1/403		H 0 4 N 1/40	1 0 3 A

審査請求 未請求 請求項の数9 O L (全 19 頁)

(21) 出願番号 特願平8-41638

(22) 出願日 平成8年(1996)2月28日

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 森 浩

東京都大田区下丸子3丁目30番2号キヤノン株式会社内

(72) 発明者 斎藤 和浩

東京都大田区下丸子3丁目30番2号キヤノン株式会社内

(72) 発明者 蒔田 剛

東京都大田区下丸子3丁目30番2号キヤノン株式会社内

(74) 代理人 弁理士 丸島 儀一

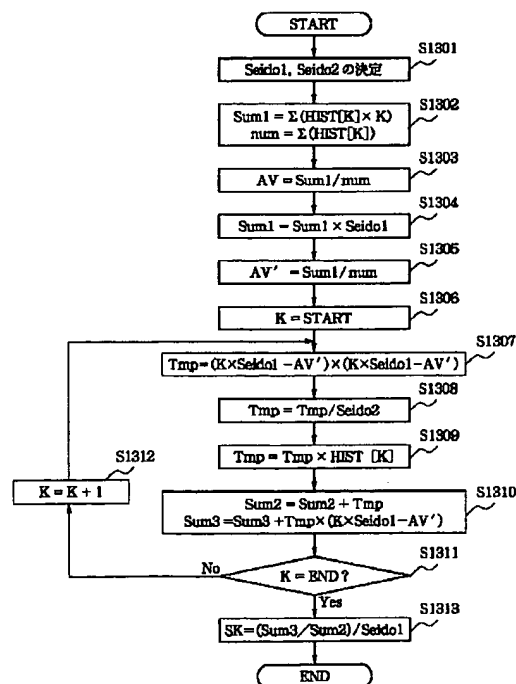
最終頁に続く

(54) 【発明の名称】 数値演算装置及びその方法及びそのコンピュータプログラム製品

(57) 【要約】

【課題】 整数演算で画像処理の数値演算を行なうと、小数部が切り捨てられ誤差が大きくなり精度が保てず、浮動小数点演算で画像処理の数値演算を行なうと精度は保てるが処理時間がかかってしまう。

【解決手段】 画像を処理する際の数値演算方法において、演算値が予め設定した有効桁数より大きくなると予測される場合は、有効桁数保存変数を用い演算値の桁数を切り捨て、演算値が予め設定した有効桁数より小さくなると予測される場合は、有効桁数保存変数を用い演算値の桁数をくり上げることにより演算値を予め設定した有効桁数内に収める。



【特許請求の範囲】

【請求項1】 画像を処理する際の数値演算方法において、演算値が予め設定した有効桁数より大きくなると予測される場合は、有効桁数保存変数を用い演算値の桁数を切り捨て、演算値が予め設定した有効桁数より小さくなると予測される場合は、有効桁数保存変数を用い演算値の桁数をくり上げることにより、演算値を予め設定した有効桁数内に収めることを特徴とする数値演算方法。

【請求項2】 有効桁数保存変数は、処理単位ごとに決定することを特徴とする請求項1記載の数値演算方法。

【請求項3】 前記処理単位は、前記画像の輝度頻度の平均値と、前記画像の輝度頻度の偏りを求める単位であることを特徴とする請求項2記載の数値演算方法。

【請求項4】 前記有効桁数は処理環境によって決定されることを特徴とする請求項1記載の数値演算方法。

【請求項5】 画像を処理する際の数値演算装置において、演算値が予め設定した有効桁数より大きくなると予測される場合に、有効桁数保存変数を用い演算値の桁数を切り捨てる第1演算手段と、演算値が予め設定した有効桁数より小さくなると予測される場合に、有効桁数保存変数を用い演算値の桁数をくり上げる第2演算手段とを有し、演算値を予め設定した有効桁数内に収めることを特徴とする数値演算装置。

【請求項6】 前記有効桁数保存変数を処理単位ごとに決定する決定手段を有することを特徴とする請求項5記載の数値演算装置。

【請求項7】 前記決定手段は前記画像の輝度頻度の平均値と、前記画像の輝度頻度の偏りを求める単位を処理単位とすることを特徴とする請求項6記載の数値演算装置。

【請求項8】 前記有効桁数は処理環境によって決定されることを特徴とする請求項5記載の数値演算装置。

【請求項9】 コンピュータプログラム製品であって、多値画像を2値画像に変換処理する数値演算を行う、コンピュータ読み取り可能なプログラムコード手段を有するコンピュータ使用可能な媒体を備え、前記プログラム製品は、演算値が処理環境によって決定される有効桁数より大きくなると予測される場合に、有効桁数保存変数を用い演算値の桁数を切り捨てるコンピュータ読み取り可能な第1プログラムコード手段と、演算値が処理環境によって決定される有効桁数より小さくなると予測される場合に、有効桁数保存変数を用い演算値の桁数をくり上げるコンピュータ読み取り可能な第2プログラムコード手段と、有効桁数保存変数を前記画像の輝度頻度の平均値と、前記画像の輝度頻度の偏りを求める単位ごとに決定するコンピュータ読み取り可能な第3プログラムコード手段と、を備えることを特徴とするコンピュータプログラム製

品。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は数値演算装置及びその方法及びそのコンピュータプログラム製品に関し、例えば、大きな値から小さい値へと数値を収束させる様な、繰り返し演算を利用した画像処理を行なう際の数値演算装置及びその方法及びそのコンピュータプログラム製品に関する。

【0002】

【従来の技術】近年の画像処理技術の高度化はめざましく、フルカラー画像等の多値画像の処理や、多値画像内の文字認識処理等が可能な画像処理など高度な数値演算処理が必要とされる画像処理装置も普及してきている。

【0003】このような画像処理技術において、多値画像の2値化処理における2値化閾値の決定においても高速、高精度の要求される数値演算処理が不可欠になっている。

【0004】従来からコンピュータ上の数値演算方法には整数演算と浮動小数点演算がある。整数演算はコンピュータの扱える整数有効数値(16bit整数で32767〜32768)内で高速に演算され、さらにコンピュータ内部の整数有効数値(CPUやシステムなどの処理環境に依存する)内では、より高速に演算される。浮動小数点演算は数値部と指数部を持った型の数値で行われるので、大きな数値から、少数点以下の数値まで表現できる。

【0005】

【発明が解決しようとしている課題】多値画像の2値化処理における2値化閾値の決定においても整数演算、浮動小数点演算を用いて画像のヒストグラムの統計的演算で2値化閾値を決定することが要求される。

【0006】そして、この2値化閾値決定の演算法においては、以下のような問題がある。

【0007】整数演算で画像処理の数値演算を行っている場合、浮動小数点演算に比べ圧倒的に高速であるが、割り算などで少数部が切り捨てられることによる誤差が発生し、繰り返し演算に割り算が含まれている場合などには誤差が大きくなりやすく精度が保てない。また整数の幅に制限があり、制限を超えた大きな数値や、小数点以下の数値は扱えない。

【0008】浮動小数点演算で画像処理の数値演算を行っている場合、高精度を保つことができるが、演算処理が遅く処理時間がかかってしまう。

【0009】本発明は上述した課題を解決するためになされたものであり、限られたコンピュータの扱える整数の有効数値内で一定の精度を保ち、特に、大きな数値から小さい数値まで収束させるような繰り返し演算の高速化を可能にする、画像処理の際の数値演算装置及びその方法及びそのコンピュータプログラム製品を提供するこ

とを目的とする。

【0010】

【課題を解決するための手段】上述した目的を達成するために、本発明の画像処理方法は以下の構成を備える。

【0011】即ち、画像を処理する際の数値演算方法において、演算値が予め設定した有効桁数より大きくなると予測される場合は、有効桁数保存変数を用い演算値の桁数を切り捨て、演算値が予め設定した有効桁数より小さくなると予測される場合は、有効桁数保存変数を用い演算値の桁数をくり上げることにより、演算値を予め設定した有効桁数内に収める。

【0012】又、本発明の画像処理装置は以下の構成を備える。

【0013】即ち、画像を処理する際の数値演算装置において、演算値が予め設定した有効桁数より大きくなると予測される場合に、有効桁数保存変数を用い演算値の桁数を切り捨てて第1演算手段と、演算値が予め設定した有効桁数より小さくなると予測される場合に、有効桁数保存変数を用い演算値の桁数をくり上げる第2演算手段とを有し、演算値を予め設定した有効桁数内に収める。

【0014】

【発明の実施の形態】以下、図面を参照して本発明に係る実施の形態について詳細に説明する。

【0015】図1は、本実施の形態における2値化処理を実行するシステム構成を示すブロック図である。図1において、1は画像の2値化処理、文字認識処理などを行う画像処理装置であり、2は画像を入力するスキャナ等の画像入力装置、3は処理後の画像を表示する若しくは処理後の画像を記録する画像出力装置である。

【0016】画像処理装置1において、4は画像入力装置2とのインターフェースとなる入力部、5は処理中のデータを記憶するメモリ等の記憶部、6は入力画像の輝度頻度（ヒストグラム）を累計する輝度頻度累計部である。7は入力画像が文字画像かどうかを判別する画像特徴判別部、8は入力画像のブロック単位の2値化閾値を算出する2値化閾値算出部であり、9はブロック単位の2値化閾値をもとに補間処理を行い画素単位の2値化閾値を算出する補間処理部、10は補間処理部9において算出された画素単位の2値化閾値を用いて2値化画像を作成する2値化部である。11は文字領域として抽出された領域に対する文字認識処理を行う文字認識部、12は文字領域以外に分離された領域に対する各種画像処理や画像のレイアウト処理を行う画像処理部、13は画像表示装置3とのインターフェースとなる出力部である。これら各構成は、CPU、ROM、RAMにより構成される制御部15により統括的に制御される。

【0017】上述した構成をなす本実施の形態の画像処理装置1において実行される文字認識処理について、以下説明する。

【0018】図2は、本実施の形態の特徴である2値化閾値決定方法を利用した文字認識処理を示すフローチャートである。

【0019】まず、ステップS201ではスキャナなどの画像入力装置2により、画像データを入力する。ここでの入力値は、8ビットの多値画像データとして行われる。続いてステップS202では、画像入力時に発生する画像入力装置2の電氣的ノイズの除去や、原稿画像の劣化、原稿の傾きなど、入力画像に対する補正を行う。続いてステップS203においては、ステップS201で入力された多値画像に対し、本実施の形態の特徴であるところの文字ブロック（つぶれ文字ブロック、かすれ文字ブロック）の判別をし、それぞれに最適な2値化閾値を決定して、該2値化閾値により2値画像を生成する。そして、ステップS204に進み、ステップS203において2値化された2値画像が、文字ブロックと判別されたブロックであるか否かの判断をし、ステップS205では、該文字ブロックを2値画像から切り出し、該2値画像に対して文字認識処理を行い、認識された文字コードを出力する。ステップS206では、未処理ブロックが画像中にあるかの判断をし、ある場合はステップS204へ、ない場合はステップS207へ進む。ステップS207では、文字ブロックと、非文字ブロックとを合成し、原稿画像と同じレイアウトにする。

【0020】《2値化処理》次に、本実施の形態の特徴であるところの文字ブロック（つぶれ文字ブロック、かすれ文字ブロック）の判別をし、2値化処理を行うステップS203について、図3のフローチャートを参照して詳細に説明する。

【0021】図3において、まずステップS301で8ビットの多値画像を画像処理装置1内の記憶部5から不図示のメモリ等に処理ブロックの単位（64画素×64画素）ごとに入力する。尚、該多値画像は、スキャナなどの画像入力装置2により、予め記憶部5に格納されているものとする。そして、ステップS302において、文字ブロック（つぶれ文字ブロック、かすれ文字ブロック）の判別をし、入力画像の処理ブロックの単位（64画素×64画素）ごとに2値化のための閾値決定処理を行う。つぶれ文字とは1文字の画数が多いため、つぶれてしまっている文字、又かすれ文字とはうすくなってしまった文字のことである。本例ではこれらの文字にも最適な閾値を割り当てることができる。ステップS303では、文字フラグMFが「1」かどうかの判断（処理中のブロックが文字ブロックかどうかの判断。詳細後述）をする。その結果が、文字ブロックであればステップS305へ、文字ブロックでなければステップS304へ進む。ステップS304では、閾値THを下限値Lと上限値Hによる制限を行う。つまり、ステップS302で決定された閾値THが、下限値Lよりも小さいときは閾値THをLで代表させ、上限値Hよりも大きい

ときは閾値THをHで代表させるような制限処理を行う。尚、この下限値Lと上限値Hは、画像入力装置2の特性により決定される値である。

【0022】ステップS305では、つぶれ文字フラグTFが「1」かどうかの判断（処理中のブロックがつぶれ文字ブロックかどうかの判断。詳細後述）をする。その結果が、つぶれ文字ブロックであればステップS306へ、つぶれ文字ブロックでなければステップS307へ進む。ステップS306では、閾値THの制限を行う。つまり、ステップS302またはステップS304で決定された閾値THに文字つぶれ防止用の定数TPをかけるような制限処理を行う。ここで、定数TPは画像入力装置2の特性により決定される値である。そして、ステップS307では、入力画像の最後の処理ブロック（64画素×64画素）かどうかの判断をし、最後の処理ブロックであればステップS308へ進み、未処理ブロックがあればS301へ戻る。ステップS308では、ステップS302、またはステップS304、またはステップS306で決定されたブロック単位の閾値THに基づきスムージング処理（詳細後述）を行い、画素単位の閾値THを算出する。最後に、ステップS309ではステップS308で算出された画素単位の閾値THを用い単純2値化処理を行い、この2値化処理を終える。

【0023】《閾値決定処理》次に、本実施の形態における2値化の閾値決定処理について、図4のフローチャートを参照して詳細に説明する。

【0024】ステップS401において、パラメータSTART、ENDにそれぞれ「0」、「255」をセットする。START、ENDはそれぞれ、後段のステップS402やステップS403で求める輝度値の統計量の始点及び終点に対応する。また、閾値決定処理ループ回数iに「0」をセットする。ステップS401では同時に、処理ブロック（64画素×64画素）中の全画素を用い、8ビット、即ち「0」から「255」までの各

$$-0.1 < SK \text{ かつ } SK < 0.1 \quad \dots (3)$$

つまり、スキュー値SKの絶対値が「0.1」未満かの判断を行う。ステップS406が真ならばステップS416へ、偽ならばステップS407へ進む。

【0033】ステップS407では、閾値決定処理ループ回数iの判断を行い、iが「1」ならばステップS408へ、「1」以外ならばステップS409へ進む。ステップS408では、処理中のブロックが「つぶれ文字ブロック」かどうかの判断と処理を行い（詳細後述）、ステップS409へ進む。ステップS409では、閾値決定処理ループ回数iの判断を行い、iが「9」ならばステップS416へ、「9」以外ならばステップS410へ進む。ステップS410では、処理中のブロックが「かすれ文字ブロック」かどうかの判断と処理を行い（詳細後述）、ステップS411へ進む。ステップS411で

デジタル値に対する頻度（ヒストグラム）を計算する。

【0025】ステップS402では、STARTからENDまでのデジタル値に対応する画素の平均値AVを算出する。たとえば、START=0、END=255であれば、「0」から「255」の値を持つ画素の平均値AVを算出し、START=0、END=109であれば「0」から「109」の値を持つ画素の平均値AVを算出する。

【0026】ステップS403では、STARTからENDまでの輝度値に対応する画素のスキュー値SKを算出する。スキュー値とは、ヒストグラム分布の偏りを示す統計量である。スキュー値の算出には、以下に示す（1）式を用いる。

$$SK = (\sum (X_i - AV)^3) / D \quad \dots (1)$$

ここで、 X_i は画素の輝度値である。また、Dは画像全体の分散値であり、（2）式により算出される。

$$D = \sum (X_i - AV)^2 \quad \dots (2)$$

【0029】上述した式（1）において、スキュー値は各画素の輝度値と、その平均値との差分を3乗することにより算出されるが、奇数乗であれば3乗に限定されるものではない。

【0030】ステップS404では、1つの処理ブロックにおける閾値決定処理ループ回数iの判断（i=0、つまり最初のループかどうかの判断）を行う。iが「0」であればステップS405へ、iが「0」以外であればステップS406へ進む。ステップS405では、処理中のブロックが「文字ブロック」かどうかの画像特徴判別を行い（詳細後述）、ステップS414へ進む。

【0031】ステップS406では（3）式に示すようにヒストグラムの偏りの大きさを判断する。

【0032】

は、かすれ文字フラグKFが「1」かどうかの判断をし、「1」ならばステップS416へ、「1」以外ならばステップS412へ進む。

【0034】ステップS412では、以下に示す（4）式によりヒストグラムの偏りの方向を判断する。

$$SK > 0 \quad \dots (4)$$

【0036】ステップS412において（4）式が真（ヒストグラムの偏りが平均値AVよりも大きい値の範囲にある事を意味する）ならばステップS413へ進む、偽（ヒストグラムの偏りが平均値AVよりも小さい値の範囲にある事を意味する）ならばステップS414へ進む。

【0037】ステップS413では、STARTに平均値AVをセットし、ENDは変化させない。そして、ス

ステップS415へ進む。ステップS414では、STARTは変化させず、ENDに平均値AVをセットする。そして、ステップS415へ進む。ステップS415では、閾値決定処理ループ回数iに「1」を加え、そしてステップS402に戻り、再びSTART値からEND値までの平均値AVを算出する。

【0038】最後に、ステップS416では平均値AVを、2値化閾値THとして設定し、この閾値決定処理を終える。

【0039】以上説明したようにして本実施の形態における2値化処理が行われるが、式(3)、(4)で示した範囲は、これに限定されるものではない。

【0040】《数値演算処理》次に本実施の形態の平均値AV算出処理(図4のS402)とスキュー値算出処理(図4のS403)の2つの処理を一連の処理としてコンピュータ上で実現した例を図17のフローチャートを参照して具体的に説明する。

【0041】ステップS1301において有効桁数保存変数Seido1、Seido2を決定する。

【0042】Seido1は整数演算で少数部分の精度を保つために演算値の桁数を繰り上げるように演算値に掛ける変数で、コンピュータ内部の整数有効数値によって決定される。本実施の形態では、コンピュータ内部整数が16bitの場合8bit分の256、32bitの場合16bit分の65536とする。本数値演算処

$$AV = (\sum (HIST[K] \times K)) / (\sum (HIST[K])) \cdots (9)$$

【0047】ステップS1302では累計変数Sum1 = $\sum (HIST[K] \times K)$ と累積変数num = $\sum (HIST[K])$ を求め、ステップS1303ではSum1をnumで割り、平均値AVを求める。

【0048】ステップS1304ではSum1にステップS1301で決定したSeido1を掛け、ステップS1305ではスキュー値算出処理用の精度確保平均値AV'を求める。

【0049】本実施の形態の式(1)(2)を演算処理しやすい形に変えたものが以下のステップS1306～S1313である。

【0050】ステップS1306はスキュー値算出処理の処理ループの輝度値Kの初期値STARTを設定する。

【0051】ステップS1307では中間変数Tmpを求めている。

【0052】 $Tmp = (K \times Seido1 - AV') \times (K \times Seido1 - AV')$ の輝度値Kには精度確保平均値AV'と同様にステップS1301で決定したSeido1を掛ける。

【0053】ステップS1308ではTmpをステップS1301で決定したSeido2で割る。例えばi=1で、64×64の処理ブロック単位、8bit画像、8bit分の精度確保の場合、次のステップで最大演算

理は多値画像の8bitデータを扱うので、元の数値部分を8bit以上確保するように桁数の繰り上げ幅(Seido1)を決めている。

【0043】Seido2は整数演算で演算結果がコンピュータの扱える整数有効数値(本実施の形態では32bit)を超える(オーバーフロー)を防ぐために演算値の桁数を切り捨てるように演算値を割る変数で、平均値AV算出とスキュー値算出を一単位とする閾値決定処理ループ回数iの数値によって決定される。閾値決定処理ループ回数を重ねるにつれSTART、END間が狭まりヒストグラムの要素数が減り、オーバーフローの可能性が少なくなるため、iの値により決定するようにしている。本実施の形態では、閾値決定処理ループ回数iの数値が大きくなるに従い、演算値の桁数切り捨て量を減少させるように以下の式によりSeido2の値を決定する。

$$Seido2 = 2^{-(10-i)} \cdots (8)$$

この値は閾値決定処理ループ回数ごとの最大演算桁数の理論上のbit数のオーバーフロー桁数により決めている。

【0045】輝度値Kにおけるヒストグラムの値をHIST[K]とすると、平均AVは以下の式で表わせる。

$$【0046】$$

桁数が理論上32bitを超え40bitになるため、このステップでオーバーフローを防ぐ処理を行う。このときSeido2の値が10なのは、処理ブロック単位に余裕を持たせるためである。

【0054】ステップS1309ではTmpにヒストグラムの値HIST[K]を掛ける。

【0055】ステップS1310では累積変数Sum2にTmpを加え、累積変数Sum3には $Tmp \times (K \times Seido1 - AV')$ を加える。

【0056】ステップS1311はK=ENDか調べ、スキュー値算出処理の処理ループ終了の判定を行う。

【0057】ステップS1312はKに1を加えステップS1307に戻る。

【0058】ステップS1313ではSum3をSum2で割ってスキュー値SKを求め、分子分母間(Sum3とSum2)での有効桁数保存変数の不均衡(分子3乗項に対し分母2乗項)を解消するため、ステップS1301で決定したSeido1で割る。

【0059】以上により数値演算処理を終わる。

【0060】以上説明した様に本実施の形態の平均値AV算出処理とスキュー値算出処理が行われるが、ここで示した有効桁数保存変数Seido1、Seido2の決定条件はこれに限定されるものではない。

【0061】以上説明したように本実施の形態における

数値演算処理によれば、コンピュータの処理環境によって決めた有効桁数保存変数を演算値に掛けることにより整数演算でも小数点以下の精度を保つことを可能にし、また平均値算出、スキュー値算出を一単位として各単位ごとに求めた有効桁数保存変数で演算値を割ることで演算結果がコンピュータの扱える整数有効数値を超えることなく演算することが可能となる。これにより、一定の精度を保ちつつ、高速な画像処理の数値演算方法を得ることができる。

【0062】《画像特徴判別処理》次に、ステップS405における画像特徴判別処理について、図5を用い詳細に説明する。ステップS501では、処理中のブロックが「文字ブロック」かどうかを示す、文字フラグMFに「0」をセットする。また、閾値決定処理ループにおける最初のスキュー値を表わすSK0に酢きゅー値SKをセットする（これは、この画像特徴判別処理は閾値決定処理ループにおける最初のループのみ処理が行われるからである）。ステップS502では、処理中のブロックが「文字ブロック」かどうかの判断を（5）式により行う。

【0063】 $SK0 < MH \quad \dots (5)$

ここで、MHは処理中のブロックが「文字ブロック」かどうかを示す値であり、ここでは、「 $MH = -20.0$ 」とする。ステップS502において（5）式が真ならばステップS503へ、偽ならばこの画像特徴判別処理を終える。ステップS503では、処理中のブロックが「文字ブロック」であることを示す、文字フラグMHに「1」をセットし、この画像特徴判別処理を終える。

【0064】以上説明したようにして本実施の形態における画像特徴判別処理が行われるが、式（5）で示した条件は、これに限定されるものではない。

【0065】《つぶれ文字処理》次に、ステップS408におけるつぶれ文字処理について、図6を用い詳細に説明する。ステップS601では、処理中のブロックが「つぶれ文字ブロック」かどうかを示す、つぶれ文字フラグTFに「0」をセットする。また、処理中のブロックのヒストグラムの偏りが大きいことを示すフラグPFに「0」をセットする。そしてステップS602で、閾値決定処理ループにおける最初のループのスキュー値SK0と、処理中のループのスキュー値SKが共にマイナスであるかの判断をし、共にマイナスであれば、ステップS603でフラグPFに「1」をセットする。ステップS604では、文字フラグMFが「1」かどうかの判断（つまり処理中のブロックが文字ブロックかどうかの判断）を行い、「1」ならばステップS605へ、「1」以外ならこのつぶれ文字処理を終える。ステップS605では、処理中のブロックが「つぶれ文字ブロック」かどうかの判断を（6）式により行う。

【0066】

$SK0/SK < -SR \quad \dots (6)$

ここで、 $-SR$ は処理中のブロックが「つぶれ文字ブロック」かどうかを示す値であり、ここでは、「 $-SR = -3.0$ 」とする。ステップS605において（6）式が真ならばステップS606へ、偽ならばこのつぶれ文字処理を終える。ステップS606では、処理中のブロックが「つぶれ文字ブロック」であることを示す、つぶれ文字フラグTFに「1」をセットし、このつぶれ文字処理を終える。

【0067】以上説明したようにして本実施の形態における、つぶれ文字処理が行われるが、式（6）で示した条件は、これに限定されるものではない。

【0068】《かすれ文字処理》さらに、ステップS410におけるかすれ文字処理について、図7を用い詳細に説明する。ステップS701では、処理中のブロックが「かすれ文字ブロック」かどうかを示す、かすれ文字フラグKFに「0」をセットする。ステップS702では、文字フラグMFが「1」かどうかの判断（処理中のブロックが文字ブロックかどうかの判断）を行い、「1」ならばステップS703へ、「1」以外ならこのかすれ文字処理を終える。ステップS703では、フラグPFが「1」かどうかの判断をし、「1」ならばステップS704へ、「1」以外ならこのかすれ文字処理を終える。ステップS704では、処理中のブロックが「かすれ文字ブロック」かどうかの判断を（7）式により行う。

【0069】

$SK0/SK > SR \quad \dots (7)$

ここで、SRは処理中のブロックが「かすれ文字ブロック」かどうかを示す値であり、ここでは、「 $SR = 3.0$ 」とする。ステップS704において（7）式が真ならばステップS705へ、偽ならばこのかすれ文字処理を終える。ステップS705では、処理中のブロックが「かすれ文字ブロック」であることを示す、かすれ文字フラグKFに「1」をセットし、このかすれ文字処理を終える。

【0070】以上説明したようにして本実施の形態におけるかすれ文字処理が行われるが、式（7）で示した条件は、これに限定されるものではない。

【0071】以下、具体的な画像の例を参照し、本実施の形態の閾値決定処理について更に詳細に説明する。

【0072】まず、図8に示すヒストグラムの例を用いて、本実施の形態における図4の2値化閾値THの決定処理について説明する。

【0073】図8は、つぶれ文字画像（8ビット入力）のヒストグラムを示したものである。図8において、横軸は、左端が「0」即ち黒、右端が「255」即ち白を表わす輝度のデジタル値であり、縦軸は、各デジタル値の頻度を表わしている。図9は、図8に示す様なヒストグラムを有する画像に対して、上述した図4で示す2値化処理においてステップS402とS403で示した処

理の際の、各パラメータの値の変化を示す図である。
尚、図9において示される各パラメータ値は、ステップS402及びS403を通過する回数によって、それぞれ示されている。

【0074】まず、ステップS402及びS403を通過する1回目の処理(閾値決定処理ループ回数 $i=0$)では、 $START=0$ 、 $END=255$ で平均値AV、統計量SKを計算し、それぞれ「109」、「-27.4」という値を得る。そして、 $i=0$ であるので、ステップS404からステップS405へ進む。ステップS405の画像特徴判別処理では、(5)式により統計量SKが「-20.0」未満であるため真となり、文字フラグMFに「1」がセットされ、ステップS414で $START=0$ 、 $END=109$ が設定される。

【0075】続いて2回目の処理($i=1$)では、 $START=0$ 、 $END=109$ における平均値AV、統計量SKを計算し、それぞれ「62」、「8.9」という値を得る。統計量SKが「0.1」を超えるため、ステップS406からステップS407へ進む。 $i=1$ であるのでステップS408へ進む。ステップS408では、スキュー値SKがプラスのため、フラグPFは「0」のままとなる。そして、(6)式の結果が「-3.08(=-27.4/8.9)」と真なので、つぶれ文字フラグTFに「1」がセットされる。そして、ステップS410では、フラグPFが「0」のままなので、かすれフラグKFは「0」のままである。さらに、スキュー値SKがプラスの為、ステップS413で $START=62$ 、 $END=109$ が設定される。

【0076】続いて3回目の処理($i=2$)では、 $START=62$ 、 $END=109$ における平均値AV、統計量DKを計算し、それぞれ「84」、「1.9」という値を得る。この場合は、統計量SKが「0.1」を超えるため、ステップS407、ステップS409へと進む。そして、スキュー値SKがプラスの為、ステップS413で $START=84$ 、 $END=109$ が設定される。

【0077】続いて4回目の処理($i=3$)では、 $START=84$ 、 $END=109$ における平均値AV、統計量SKを計算し、それぞれ「96」、「0.6」という値を得る。この場合は、統計量SKが「0.1」を超えるため、処理が続き、スキュー値SKがプラスの為、ステップS413で $START=96$ 、 $END=109$ が設定される。

【0078】続いて5回目の処理($i=4$)では、 $START=96$ 、 $END=109$ における平均値AV、統計量SKを計算し、それぞれ「102」、「-0.3」という値を得る。この場合は、統計量SKが「-0.1」未満のため、処理が続き、スキュー値SKがマイナスの為、ステップS414で $START=96$ 、 $END=102$ が設定される。

【0079】続いて6回目の処理($i=5$)では、 $START=96$ 、 $END=102$ における平均値AV、統計量SKを計算し、それぞれ「98」、「0.3」という値を得る。この場合は、統計量SKが「0.1」を超えるため、処理が続き、スキュー値SKがプラスの為、ステップS413で $START=98$ 、 $END=102$ が設定される。

【0080】続いて7回目の処理($i=6$)では、 $START=98$ 、 $END=102$ における平均値AV、統計量SKを計算し、それぞれ「99」、「0.4」という値を得る。この場合は、統計量SKが「0.1」を超えるため、処理が続き、スキュー値SKがプラスの為、ステップS413で $START=99$ 、 $END=102$ が設定される。

【0081】続いて8回目の処理($i=7$)では、 $START=99$ 、 $END=102$ における平均値AV、統計量SKを計算し、それぞれ「100」、「0.2」という値を得る。この場合は、統計量SKが「0.1」を超えるため、処理が続き、スキュー値SKがプラスの為、ステップS413で $START=100$ 、 $END=102$ が設定される。

【0082】続いて9回目の処理($i=8$)では、 $START=100$ 、 $END=102$ における平均値AV、統計量SKを計算し、それぞれ「101」、「-0.2」という値を得る。この場合は、統計量SKが「-0.1」未満のため、処理が続き、スキュー値SKがマイナスの為、ステップS414で $START=100$ 、 $END=101$ が設定される。

【0083】続いて10回目の処理($i=9$)では、 $START=100$ 、 $END=101$ における平均値AV、統計量SKを計算し、それぞれ「100」、「0.2」という値を得る。この場合は、統計量SKが「0.1」を超えるが、閾値決定処理ループ回数 i が「9」のため、ステップS409における判断により、ステップS416へと進む。閾値THに平均値AVである「100」をセットし、この閾値決定処理を終える。ただし、この例におけるブロックは、つぶれ文字フラグTFに「1」がセットされているため、図3のステップS305の判断が真となり、ステップS306で閾値に制限を受け、THには「85(=100×0.85)」がセットされ、ステップS308での閾値THのスムージング処理の後、ステップS309で2値化される。(この場合TPに「0.85」がセットされている。)そして、2値化された画像は記憶部5に格納される。

【0084】次に、図10に示すヒストグラムの例を用いて、本実施の形態における図4の2値化閾値THの決定処理について説明する。

【0085】図10は、かすれ文字画像(8ビット入力)のヒストグラムを示したものである。図11は、図10に示す様なヒストグラムを有する画像に対して、上

述した図4で示す2値化処理においてステップS402とS403で示した処理の際の、各パラメータの値の変化を示す図である。尚、図11において示される各パラメータ値は、ステップS402及びS403を通過する回数によって、それぞれ示されている。

【0086】まず、ステップS402及びS403を通過する1回目の処理(閾値決定処理ループ回数 $i=0$)では、 $START=0$ 、 $END=255$ で平均値 AV 、統計量 SK を計算し、それぞれ「130」、「-60.2」という値を得る。そして、 $i=0$ であるので、ステップS404からステップS405へ進む。ステップS405の画像特徴判別処理では、(5)式により統計量 SK が「-20.0」未満であるため真となり、文字フラグ MF に「1」がセットされ、ステップS414で $START=0$ 、 $END=130$ が設定される。

【0087】続いて2回目の処理($i=1$)では、 $START=0$ 、 $END=130$ における平均値 AV 、統計量 SK を計算し、それぞれ「95」、「-19.3」という値を得る。統計量 SK が「0.1」を超えるため、ステップS406からステップS407へ進む。 $i=1$ であるのでステップS408へ進む。ステップS408では、スキュー値 SK 、 $SK0$ が共にマイナスのため、フラグ PF に「1」がセットされる。そして、(6)式の結果が「3.11($=-60.2/-19.3$)」と偽なので、つぶれ文字フラグ TF は「0」のままとなる。そして、ステップS410のかすれ文字処理では、文字フラグ MF 、フラグ PF 共に「1」なので、ステップS704で(7)式の計算を行い、結果が「3.11($=-60.2/-19.3$)」と真なので、かすれ文字フラグ KF に「1」をセットする。かすれフラグ KF が「1」なので、ステップS411からステップS416へと進み、閾値 TH に平均値 AV の「95」をセットし、この閾値決定処理を終える。そして、図3のステップS308での閾値 TH のスムージング処理の後、ステップS309で2値化され、2値化された画像は記憶部5に格納される。

【0088】《スムージング処理》では、図3のステップS308のスムージング処理の説明を図12～図16を用いて説明する。

【0089】図12は、図3のステップS308のスムージング処理の構成を最も良く示すフローチャートである。図12において、ステップS1201からステップS1204までは、ブロック単位で行なうスムージング処理の各種パラメータの設定を行なうステップであり、ステップS1201は、スムージングを行なうブロックの垂直方向スムージング領域の設定、ステップS1202は、スムージングを行なうブロックの水平方向スムージング領域の設定、ステップS1203は、スムージングを行なうブロックの垂直方向スムージングステップの設定、ステップS1204は、スムージングを行なうブ

ロックの水平方向のスムージングステップの設定を行なう。そして、ステップS1205からステップS1209までは、ステップS1201からステップS1204までの設定に基づきブロック単位で2値化閾値のスムージングを実行するステップである。ステップS1205は、スムージング処理ブロック画像の抽出を行ない、ステップS1206は、スムージングを行なうブロックの4隅の閾値の設定、ステップS1207は、垂直方向スムージング処理、ステップS1208は、水平方向スムージング処理を行なう。そして、ステップS1209では未処理のスムージングを行なうブロックがあるか否かを判断し、未処理のブロックがある場合は、ステップS1205に行き、スムージング処理を継続し、未処理のスムージングを行なうブロックがない場合は、終了する。

【0090】以下、各ステップの詳細を行なうために、ステップS301～S307までのループにおけるブロック単位の閾値を算出するブロックとスムージング処理を行なうブロックとの関係について説明したあと、ステップS1201～ステップS1209までの各ステップの詳細を説明する。

【0091】図13は、図3のステップS301～S307までのループにおけるブロック単位の閾値を算出するブロックの単位を示しているもので、左上隅から 64×64 画素毎にブロックが区切られており、 $B11 \sim Bnm$ とブロックに番号が示されている。以下、ステップS301～S307までのブロック単位の閾値の算出ループをブロック閾値算出ループと呼び、その結果得られた閾値をブロック閾値と呼ぶ。

【0092】図14は、ブロック閾値算出ループにて算出された2値化のためのブロック閾値と画素単位の閾値を算出するためのスムージング処理の単位となるブロックとの関係を示した図である。図14において、細い実線は図13のブロック分けと同様のもので、左上第1画素を起点とした 64×64 のブロック分けを示す。ブロック閾値算出ループでは、ブロック閾値はこの細い実線で分割されたブロック単位で算出されるので、ブロック閾値はそのブロックを代表する閾値と考えることができる。そこで、この細い実線で分割されたブロックの中心点、即ち、太い点線の交点がブロック閾値算出ループより得られたブロック閾値とし、ステップS308のスムージング処理では、その新たに太い点線で分割されたブロック単位で、各太い点線で分割されたブロックの4隅の閾値に基づき補間処理することにより、その太い点線で分割されたブロックの画素単位の閾値を算出する。以下、この太い点線で分割されたブロックをスムージングブロックと呼ぶ。

【0093】つまり、ステップS308のスムージング処理では、図14における太い点線で分割された $BL1$ 、 $BL12$ 、...、 $BL1(m-1)$ 、 $BL1m$ 、 B

L21, b122, . . . BLn (m-1), BLnm
を新たにスムージング処理のためのスムージングブロックと定義し、各ブロックの交点、即ち、太い点線の交点には、図中に示しているようなブロック閾値を定義する。例えば、BL11の4隅の閾値は、TH11, TH11, TH11, TH11であり、BL22の4隅の閾値は、TH11, TH12, TH21, TH22である。そして、スムージングブロックにおいて、その4隅の閾値により、垂直方向、又は、水平方向の線形補間を行なう。たとえば、BL22のスムージング処理は、TH11, TH12, TH21, TH22の4つの閾値により垂直水平方向の線形補間がおこなわれる。

【0094】さて、図12のステップS1201からS1204までを、図15を用いて説明する。

【0095】図15において、TH_TL, TH_TR, TH_BL, TH_BRは、そのスムージングブロックにおける4隅のブロック閾値を示し、各スムージングブロックにおけるTH_TL, TH_TR, TH_BL, TH_BRの値とブロック閾値算出ループより算出されたブロック閾値との関係は、前記に説明した通りである。

【0096】ステップS1201は、垂直方向スムージング領域の設定を行なう。垂直方向スムージング領域とは、垂直方向の補間演算を行なう領域であり、図15に示されているように、V1とV2によって設定されている。例えば、V1=1, V2=64の場合、垂直方向補間演算は全領域に対して行なわれる、また、V1=1, V2=48の時は、スムージングブロックにおいて、第1行から第16行までは、垂直方向の補間演算が行なわれず第1行の閾値が繰り返し用いられる、第17行から第48行までは、TH_TL, TH_TR, TH_BL, TH_BRを用いて垂直方向の補間演算が行なわれ、そして、第49行から第64行までは、垂直方向の補間演算が行なわれず、第49行の値が繰り返し用いられる。

【0097】ステップS1202は、水平方向のスムージング領域の設定を行なう。水平方向のスムージング領域とは、水平方向の補間演算を行なう領域であり、図15に示されているように、H1とH2によって設定される。例えば、H1=1, H2=64の場合、水平方向補間演算は全領域に対して行なわれる、また、H1=1, H2=48の時は、スムージングブロックにおいて、第1列から第16列までは、水平方向の補間演算が行なわれず第1列の閾値が繰り返し用いられる、第17列から第48列までは、TH_TL, TH_TR, TH_BL, TH_BRを用いて水平方向の補間演算が行なわれ、そして、第49列から第64列までは、水平方向の補間演算が行なわれず、第49列の値が繰り返し用いられる。

【0098】従って、図15において、(0, 0)を左

上隅、(H1, V1)を右下隅とした四角形領域は、TH_TLの値が繰り返され、(H2, 0)を左上隅、(64, V1)を右下隅とした四角形領域は、TH_TRの値が繰り返され、(0, V2)を左上隅、(H1, 64)を右下隅とした四角形領域は、TH_BLの値が繰り返され、(H2, V2)を左上隅、(64, 64)を右下隅とした四角形領域は、TH_BRの値が繰り返される。(0, V1)を左上隅、(H1, V2)を右下隅とした四角形領域と(H2, V1)を左上隅、(64, V2)を右下隅とした四角形領域は、垂直方向の補間演算のみが行われ、(H1, 0)を左上隅、(H2, V1)を右下隅とした四角形領域と(H1, V2)を左上隅、(H2, 64)を右下隅とした四角形領域は、水平方向の補間演算のみが行われる。そして、(H1, V1)を左上隅、(H2, V2)を右下隅とした四角形領域は、垂直方向と水平方向の両方の補間演算が行われる。

【0099】以上説明したように、垂直、水平方向のスムージング領域を小さくすればするほど、スムージングのための補間演算回数が減り、スムージング処理のための計算時間を短くすることができる、しかし、スムージング領域を小さくすればするほど、ブロックの境界の画像データの変化が急峻となるため、ブロック歪みが出やすくなるという特性を持つ。本実施の形態では、この領域設定が可変であるため、入力装置や入力画像の解像度に最適なスムージング領域の設定が可能となる。

【0100】ステップS1203は、垂直方向スムージングステップの設定を行なう。垂直方向スムージングステップとは、ステップS1201で設定された垂直方向のスムージング領域内の補間演算を何画素おきに演算するかを示す値であり、VSTEPによって設定される。例えば、垂直方向スムージングステップVSTEPが1の時は、1画素毎に補間演算を行ない、4の時は、4画素おきに行なう。そして、複数画素おきに補間演算を行なった場合、その補間演算により得られた値と値の間の値は、補間演算によって得られた値を繰り返して用いる。例えば、V1=1, V2=64で垂直方向スムージングステップVSTEPが16の場合、第1行、第17行、第33行、第49行の値が、4隅のブロック閾値から補間演算によって算出される。そして、第2行から16行までは、第1行の値が繰り返され、第18行から第32行までの値は、第17行の値が繰り返され、第34行から第48行までの値は、第33行の値が繰り返され、第50行から第64行までの値は、第49行の値が繰り返される。

【0101】ステップS1204は、水平方向スムージングステップの設定を行なう。水平方向スムージングステップとは、ステップS1202で設定された水平方向のスムージング領域内の補間演算を何画素おきに演算するかを示す値であり、HSTEPによって設定される。

例えば、水平方向スムージングステップHSTEPが1の時は、1画素毎に補間演算を行ない、4の時は、4画素おきに行なう。そして、複数画素おきに補間演算を行なった場合、その補間演算により得られた値と値の間の値は、補間演算によって得られた値を繰り返して用いる。例えば、 $H1=1$ 、 $H2=64$ で水平方向スムージングステップHSTEPが16の場合、第1列、第17列、第33列、第49列の値が、4隅のブロック閾値から補間演算によって算出される。そして、第2列から第16列までは、第1列の値が繰り返され、第18列から第32列までの値は、第17列の値が繰り返され、第34列から第48列までの値は、第33列の値が繰り返され、第50列から第64列までの値は、第49列の値が繰り返される。

【0102】従って、垂直、水平方向のスムージングステップを小さくすればするほど、スムージングのための補間演算回数が減り、スムージング処理のための計算時

$$\begin{aligned} VL_i &= TH_TL + (i-1) \cdot (TH_BL - TH_TL) / ((V2 - V1 + 1) / VSTEP) \\ &\dots (12-1) \\ VR_i &= TH_TR + (i-1) \cdot (TH_BR - TH_TR) / ((V2 - V1 + 1) / VSTEP) \\ &\dots (12-2) \end{aligned}$$

と表すことができる。そして、この VL_i と VR_i を用いて各行の水平方向補間処理を行なう。その補間結果を Hi_j ($i=1, 2, \dots, (V2-V1+1)/V$

$$Hi_j = VL_i + (j-1) \cdot (VR_i - VL_i) / ((H2 - H1 + 1) / HSTEP) \dots (12-3)$$

と表すことができる。ここで、 $(V2-V1+1)/VSTEP$ は、垂直方向の1ライン当たりの補間演算回数を、 $(H2-H1+1)/HSTEP$ は、水平方向の1ライン当たりの補間演算回数を示す。

【0104】 $V1=1$ 、 $V2=64$ 、 $H1=1$ 、 $H2=64$ 、 $VSTEP=1$ 、 $HSTEP=1$ で、 $TH_TL=32$ 、 $TH_TR=96$ 、 $TH_BL=96$ 、 $TH_BR=160$ とすると、上記(12-1)式、(12-2)式、(12-3)式からスムージングブロック内の閾値を算出すると図16のようになる。

【0105】次に、ステップS1205からステップS1209までのループの処理について詳細に説明する。このループは、図14に示した $BL11$ 、 $BL12$ 、 \dots 、 $BL1(m-1)$ 、 $BL1m$ 、 $BL21$ 、 $BL22$ 、 \dots 、 $BLn(m-1)$ 、 $BLnm$ のスムージングブロック毎に処理するループである。

・ステップS1205において、 $BL11$ の画像データを抽出する。

・ステップS1206にて、その4隅の閾値が、 $TH_TL=TH11$ 、 $TH_TR=TH11$ 、 $TH_BL=TH11$ 、 $TH_BR=TH11$ のように設定される。

・ステップS1207では、垂直方向のスムージングス

間を短くすることができる。しかし、スムージングステップを小さくすればするほど、スムージング領域の閾値の変化が粗くなるため、ブロック歪みが出やすくなるという特性を持つ。本実施の形態では、この領域設定が可変であるため、入力装置や入力画像の解像度に最適なスムージングステップの設定が可能となる。

【0103】次に、ステップS1205からステップS1209までの補間演算を行なうループの説明を行なう前に、図15を用いて、垂直方向と水平方向の補間演算方法を説明した後で、補間演算ループの説明を行なう。図15に於いて、垂直方向の補間演算に関し、 TH_TL と TH_BL を用いた垂直方向補間処理結果を VL_i ($i=1, 2, \dots, (V2-V1+1)/VSTEP$)とし、 TH_TR と TH_BR を用いた垂直方向補間処理結果を VR_i ($i=1, 2, \dots, (V2-V1+1)/VSTEP$)とすると、 VL_i 、 VR_i は、

$$STEP, j=i=1, 2, \dots, (H2-H1+1)/HSTEP)とすると、 Hi_j は、$$

理を行なうが、 TH_TL と TH_BL の値が同じで、 TH_TR と TH_BL の値が同じため、垂直方向の補間演算は行なわれず、同じ値 $TH11$ が繰り返される。

・ステップS1208では、水平方向の補間処理に関しても左端と右端の値が同じため、水平方向の補間演算は行なわれず、同じ値 $TH11$ が繰り返される。

・ステップS1209では、未処理ブロックがあるのでステップS1205に進む。

【0106】次にスムージングブロック $BL12$ の処理が行なわれる。

・ステップS1205では、次のスムージングブロック $BL12$ の画像データが抽出される。

・ステップS1206では、4隅のブロック閾値 $TH_TL=TH11$ 、 $TH_TR=TH12$ 、 $TH_BL=TH11$ 、 $TH_BR=TH12$ が設定される。

・ステップS1207では、 TH_TL と TH_BL 、そして、 TH_TR と TH_BR が同じ値のため、垂直方向の補間演算は行なわれず、 VL_i では $TH11$ が、 VR_i では $TH12$ が繰り返される。

・ステップS1208では、上記(12-3)式に基づき、水平方向の補間演算が行なわれる。ただし、 $i=1, 2, \dots, (V2-V1+1)/VSTEP$ にお

いて、常に $VL_i = TH_{11}$ 、 $VR_i = TH_{12}$ であるため、 H_{ij} の値は、 i が変化しても同じとなる。従って、 $i=2$ 以降は、 $i=1$ の時の値 H_{1j} ($j=1, 2, \dots, (H_2-H_1+1)/HSTEP$) が繰り返し用いられる。

・ステップS1209では、未処理ブロックがあるのでステップS1205に進む。

以下、上記スムージングブロックBL12と同様の処理がスムージングブロックBL13, BL14, ..., BL1mまで繰り返される。

【0107】次にスムージングブロックBL21の処理が行なわれる。

・ステップS1205では、次のスムージングブロックBL21の画像データが抽出される。

・ステップS1206では、4隅のブロック閾値 $TH_TL = TH_{11}$ 、 $TH_TR = TH_{11}$ 、 $TH_BL = TH_{21}$ 、 $TH_BR = TH_{21}$ が設定される。

・ステップS1207では、上記(12-1)式に基づき VL_i が、上記(12-2)式に基づき VR_i の補間演算が行なわれ、 VL_i 、 VR_i が求められる。

・ステップS1208では、 TH_TL と TH_TR 、そして、 TH_BL と TH_BR が同じため VL_i と VR_i の値が、 $i=1, 2, \dots, (V_2-V_1+1)/VSTEP$ の全てにおいて、同じとなるため $i=1$ の時のみ、水平方向の補間演算は行なわれず、 VL_i の値が H_{1j} ($j=1, 2, \dots, (H_2-H_1+1)/HSTEP$) が繰り返し用いられる。そして、この繰り返し、 $i=1, 2, \dots, (V_2-V_1+1)/VSTEP$ まで行なわれる。

・ステップS1209では、未処理ブロックがあるのでステップS1205に進む。

【0108】次にスムージングブロックBL22の処理が行なわれる。

・ステップS1205では、次のスムージングブロックBL22の画像データが抽出される。

・ステップS1206では、4隅のブロック閾値 $TH_TL = TH_{11}$ 、 $TH_TR = TH_{12}$ 、 $TH_BL = TH_{21}$ 、 $TH_BR = TH_{22}$ が設定される。

・ステップS1207では、上記(12-1)式に基づき VL_i が、上記(12-2)式に基づき VR_i の補間演算が行なわれ、 VL_i 、 VR_i が求められる。

・ステップS1208では、上記(12-3)式に基づき、水平方向の補間演算が行なわれ、 H_{ij} が求められる。

・ステップS1209では、未処理ブロックがあるのでステップS1205に進む。

【0109】以下、上記と同様の処理がBLnmまで繰り返されて、画像全域の2値化閾値のステップS308のスムージング処理が終了する。

【0110】この様に本実施の形態の閾値決定方法によ

れば、多値画像中のブロックごとに輝度頻度を算出し、該輝度頻度に基づいてブロック単位の2値化閾値を算出し、該ブロック単位の2値化閾値を補間することで画素単位の2値化閾値を算出することにより、即ち、画像中のブロック単位で対象物と背景濃度の間に適切な閾値を自動的に設定し、さらに画素単位で2値化閾値を自動的に設定することが可能になる。これにより、画像中の文字部分/それ以外の部分それぞれに対し、背景から対象物を適切に分離し、かつブロック歪の無い2値化画像を得ることができる。

【0111】《スムージング処理の他の例》前記実施の形態では、スムージング処理を実施するために、垂直方向のスムージング処理を行った後に水平方向のスムージング処理を行っているが、この順番はこれに限らず、水平方向のスムージング処理の後で垂直方向のスムージング処理を実施してもよい。

【0112】また、前記実施の形態では、スムージング処理を実施するためスムージングブロックを定義し、スムージングブロックごとに補間演算を行なっているが、この演算順序はこれに限らず、上下左右、様々に分割して、スムージング処理を実現しても良い。例えば、全画像領域の第1ラインの補間演算を行なった後に、第2ライン、第3ラインとラスタ順次で行っても良い。さらに、実施の形態で定義したスムージングブロックのBL21, BL22, ..., BL2m, BL31, ..., BLnmまでを上下2つに分けて、BL11, BL12, ..., BL1m
BL21上, BL22上, ..., BL2m上
BL21下, BL22下, ..., BL2m下
BL31上, BL32上, ..., BL3m上
BL31下, BL32下, ..., BL3m下
.....

BLn1下, BLn2下, ..., BLnm下
とスムージングをすることもできる。

【0113】このとき、スムージングパラメータを $V_1 = 33$ 、 $V_2 = 64$ 、 $H_1 = 1$ 、 $H_2 = 64$ と設定し、スムージングブロックの第1行から第32行までと第33行からスムージングブロックの最終行までとで上下に2つに分けると、スムージングブロックの上側は、 TH_TL と TH_TR から水平方向のスムージングのみを行ない、下側は TH_TL 、 TH_TR 、 TH_BL 、 TH_BR の4つから垂直水平方向のスムージングを行なう。この場合は、BL11, BL12, ..., BL1m, BL21上, ..., BL2m上は、図13の第1バンドと第2バンドの結果の閾値のみを用いてスムージング処理をすることができ、BL21下, BL22下, ..., BL2m下, BL3上, BL32上, ..., BL3m上は、図13の第1バンドと第2バンドの結果の閾値のみを用いてスムージング処理をすることができ、BL31下以降も、スムージングプロ

ックの下側と上側をセットにして処理することにより、前バンドのブロック閾値を保存しておくだけで、図13に示されているバンド単位で処理することができる。これにより、本発明をバンド処理で実現する場合に複数バンド持つことなく、シングルバンドでスムージング処理を実現できるため、画像用バンドメモリを節約することができる。

【0114】(他の実施の形態)前記実施の形態において、画像の入力は、1画素単色の8ビットの多値画像データとしたが、これに限定する必要はなく、複数のカラーであってもよい。つまりは、2値化するためのデータとして複数ビットの情報があればよい。

【0115】また、ヒストグラムを算出する際の画像の処理ブロックについて、64画素×64画素としたが、ブロックの大きさは32画素×32画素でもよい。さらにはブロックの形は正方形でも長方形でもよい。つまり、処理ブロックを64画素×96画素のようにしてもよい。メモリの容量によっては、画像全面を処理ブロックとしてもよい。

【0116】また、ヒストグラムを算出する際の画像におけるサンプリングについて、全画素でも、数画素おきでもよい。さらに、平均AVや統計量SKなどの計算は、8ビットで行なわなくてもよく、高速化、メモリ削減等のため、少ないビット数で演算するようにしてもよい。

【0117】また、統計量SKの収束条件を「 ± 0.1 」としたが、これに限定することではなく、別の条件を定めてもよく、統計量SKを用いて2値化の閾値を決定するように構成すればよく、限定しない。また、画像入力装置や画像入力条件により変化させる等、別の条件を定めてもよい。

【0118】また、(5)式の文字ブロックの判定に関し、文字フラグMFを「 -20.0 」としたが、これに限定することではなく、統計量SKを用いて文字ブロックの判定を行うよう、別の条件を定めてもよい。また、画像入力装置や画像入力条件により変化させる等、別の条件を定めてもよい。

【0119】また、(6)式、(7)式でつづれ文字ブロックやかすれ文字ブロックの判定に、SR値を用い「 3.0 」としたが、これに限定することではなく、統計量SKを用いてつづれ文字ブロックやかすれ文字ブロックの判定を行うよう、別の条件を定めてもよい。また、画像入力装置や画像入力条件により変化させる等、別の条件を定めてもよい。

【0120】また、閾値の下限値Lと上限値Hについて、画像入力装置や画像入力条件により変化させる等、構成してもよい。

【0121】また、つづれ防止の為に定数TPを用い「 0.85 」としたが、これに限定することではなく、画像入力装置や画像入力条件により変化させる等、別の条

件を定めてもよい。

【0122】また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記憶した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ(またはCPUやMPU)が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。

【0123】この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0124】プログラムコードを供給するための記憶媒体としては、例えば、フロッピーディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROMなどを用いる事ができる。

【0125】例えば、図18を例に説明すると、1~3は図1の1~3に対応し、この装置のCPU101が、フロッピーディスク103に格納されたプログラムコードをメモリ102に読み出し、実行することによって達成される。

【0126】また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS(オペレーティングシステム)などが実際の処理の一部又は全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0127】さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0128】本発明を上記記憶媒体に適応する場合、その記憶媒体には、先に説明したフローチャートに対応するプログラムコードを格納することになるが、簡単に説明すると、図19のメモリマップ例に示す各モジュールを記憶媒体に格納することになる。

【0129】すなわち、例えば、図3のS301処理ブロック抽出処理に対応する「ブロック処理モジュール」、図4のS401ヒストグラム算出処理に対応する「ヒストグラム算出モジュール」、図4のS402平均値算出処理に対応する「平均値算出モジュール」、図4のS403SK値算出処理に対応する「Skew値算出モジュール」、図4のS405画像特徴判別処理に対応する「画像特徴判別処理モジュール」、図4のS416

閾値算出処理、または図3のS304閾値制限処理、または図3のS306閾値制限処理に対する「閾値算出モジュール」、図3のS308スムージング処理に対応する「スムージング処理モジュール」および、図3のS309単純2値化処理に対応する「2値化処理モジュール」、の各モジュールのプログラムコードを記憶媒体に格納すればよい。

【0130】以上説明したように、本実施例の形態によれば、2値化においてブロック歪のない滑らかな2値再現が可能となる。

【0131】特に、本実施の形態によれば、多値画像中のブロックごとに輝度頻度を算出し、該輝度頻度に基づいてブロック単位の2値化閾値を算出し、該ブロック単位の2値化閾値を補間することで画素単位の2値化閾値を算出することにより、即ち、画像中のブロック単位で対象物と背景濃度の間に適切な閾値を自動的に設定し、さらに画素単位で2値化閾値を自動的に設定することが可能になる。これにより、画像中の文字部分／それ以外の部分それぞれに対し、背景から対象物に適切に分離し、かつブロック歪の無い2値化画像を得ることができ

る。

【0132】また、本実施の形態の補間（スムージング処理）により、ブロック歪みを除去することができる。また、スムージングのためのパラメータを調整することにより、スムージングのための補間処理時間を短縮したり、入力装置や入力画像の解像度等の特性に応じた補間処理を実現することができる。さらに、ブロックと同じ高さの画像用バンドメモリを用いたバンド処理を本発明に適用した場合、スムージングパラメータの設定により、シングルバンドのみでスムージングを実現することができ、バンドメモリに必要となるコストを抑えることができる。

【0133】又、コンピュータの処理環境によって決めた有効桁数保存変数を演算値に掛けることにより整数演算でも小数点以下の精度を保つことを可能にし、また平均値算出、スキュー値算出を一単位として各单位ごとに求めた有効桁数保存変数で演算値を割ることで演算結果がコンピュータの扱える整数有効数値を超えることなく演算することが可能となる。これにより、一定の精度を保ちつつ、高速な画像処理の数値演算方法を得ることができる。

【0134】

【発明の効果】以上説明したように本発明によれば、コンピュータの処理環境によって決めた有効桁数保存変数を演算値に掛けることにより整数演算でも少数点以下の精度を保つことを可能にし、また平均値算出、スキュー値算出を一単位として各单位ごとに求めた有効桁数保存変数で演算値を割ることで演算結果がコンピュータの扱える整数有効数値を超えることなく演算することが可能となる。これにより、一定の精度を保ちつつ、高速な画

像処理の数値演算を行う事ができる。

【図面の簡単な説明】

【図1】本発明に係る実施の形態における画像処理装置のシステム構成を示すブロック図である。

【図2】本実施の形態における文字認識処理を示すフローチャートである。

【図3】本実施の形態における2値化処理を示すフローチャートである。

【図4】本実施の形態における閾値決定処理を示すフローチャートである。

【図5】本実施の形態における画像特徴判別処理を示すフローチャートである。

【図6】本実施の形態におけるつぶれ文字処理を示すフローチャートである。

【図7】本実施の形態におけるかすれ文字処理を示すフローチャートである。

【図8】本実施の形態におけるつぶれ文字画像のヒストグラム例を示す図である。

【図9】本実施の形態のつぶれ文字の2値化処理における各変数値の変遷例を示す図である。

【図10】本実施の形態におけるかすれ文字画像のヒストグラム例を示す図である。

【図11】本実施の形態のかすれ文字の2値化処理における各変数値の変遷例を示す図である。

【図12】本実施の形態におけるスムージング処理を示すフローチャートである。

【図13】図3のステップS301～ステップS307までのブロック単位の閾値を算出するループにおけるブロックを示す図である。

【図14】スムージングブロックとブロック閾値との関係を示す図である。

【図15】スムージングブロックの4隅の閾値と垂直方向スムージング領域、水平方向スムージング領域を示す図である。

【図16】 $TH_TL=32$ 、 $TH_TR=96$ 、 $TH_BL=96$ 、 $TH_BR=160$ としたときの、スムージングブロック内の補間演算結果を示す図である。

【図17】数値演算処理を示すフローチャートである。

【図18】本発明に係る実施の形態におけるコンピュータプログラム製品例を説明する図である。

【図19】本発明の実施形態である画像処理を実行するためのプログラムモジュールの所定の記憶媒体でのレイアウトの一例の図である。

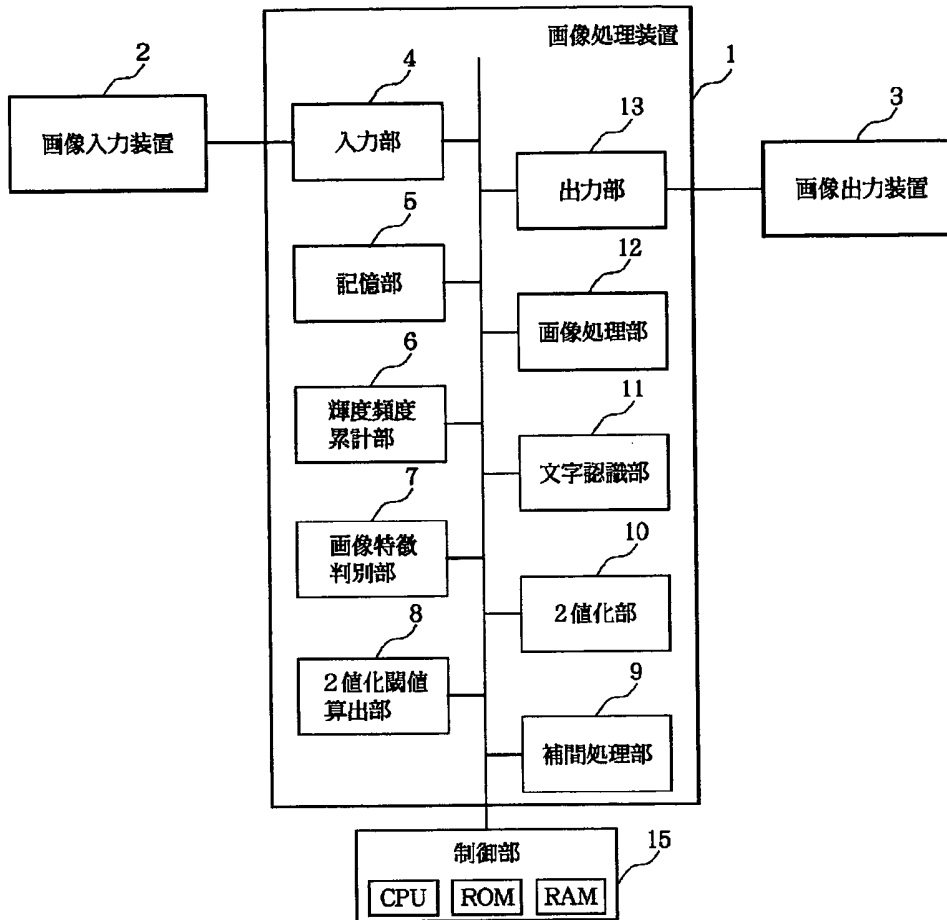
【符号の説明】

- 1 画像処理装置
- 2 画像入力装置
- 3 画像表示装置
- 4 入力部
- 5 記憶部
- 6 輝度頻度累計部

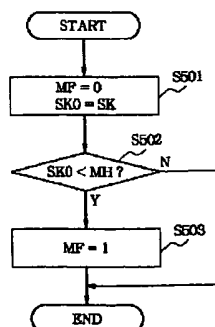
- 7 画像特徴判別部
8 2値化閾値算出部
9 補間処理部
10 2値化部

- 11 文字認識部
12 画像処理部
13 出力部
15 制御部

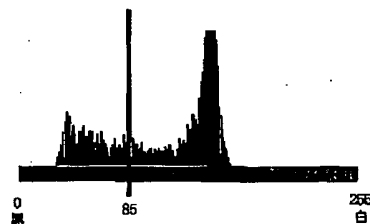
【図1】



【図5】



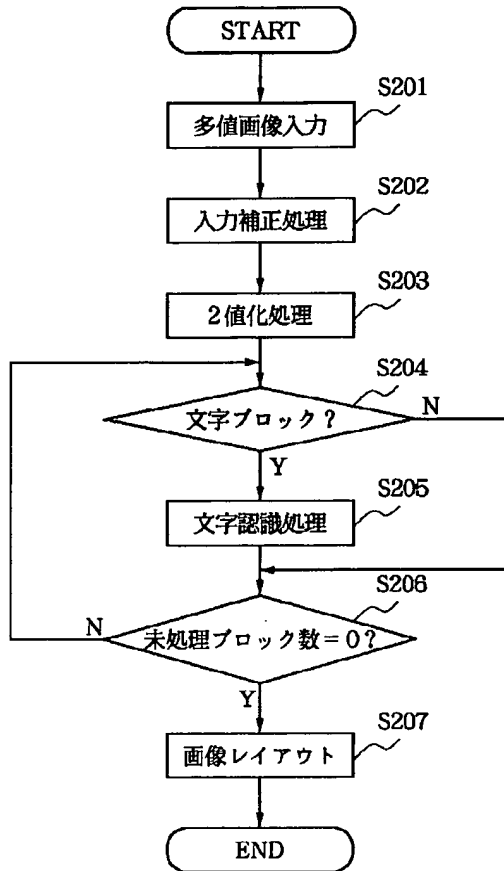
【図8】



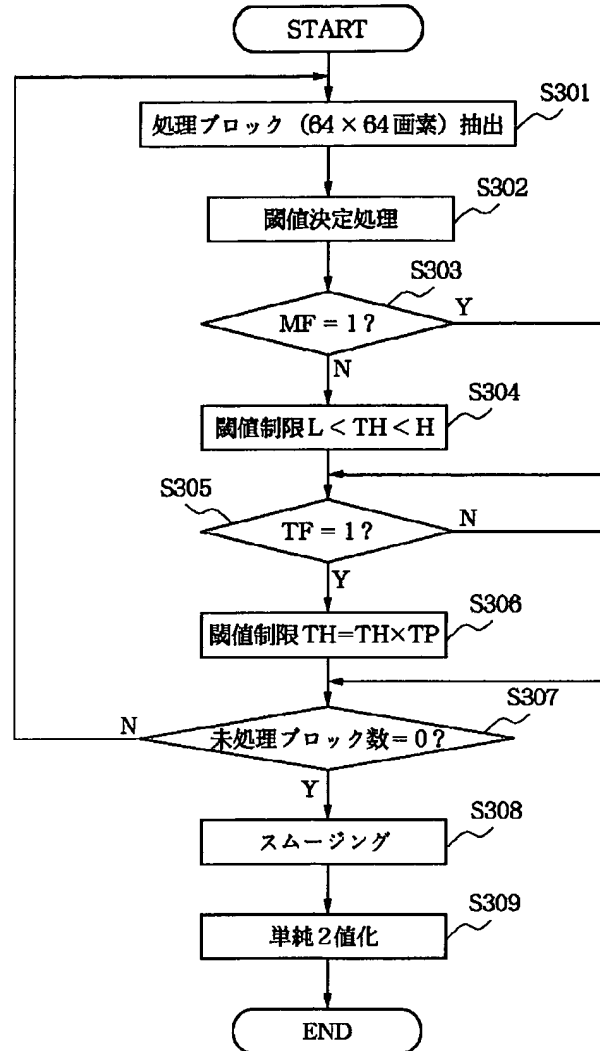
【図9】

	START	END	AV	SK
1回目	0	255	109	-27.4
2回目	0	109	62	8.9
3回目	62	109	84	1.9
4回目	84	109	98	0.8
5回目	98	109	102	-0.3
6回目	98	102	98	0.3
7回目	98	102	99	0.4
8回目	99	102	100	0.2
9回目	100	102	101	-0.2
10回目	100	101	100	0.2

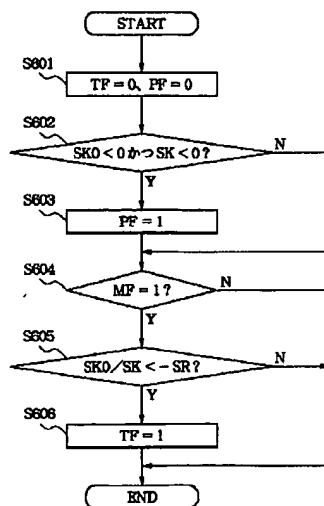
【図2】



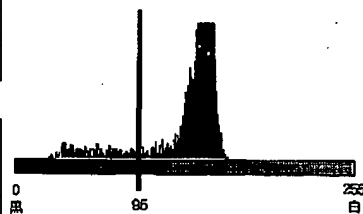
【図3】



【図6】



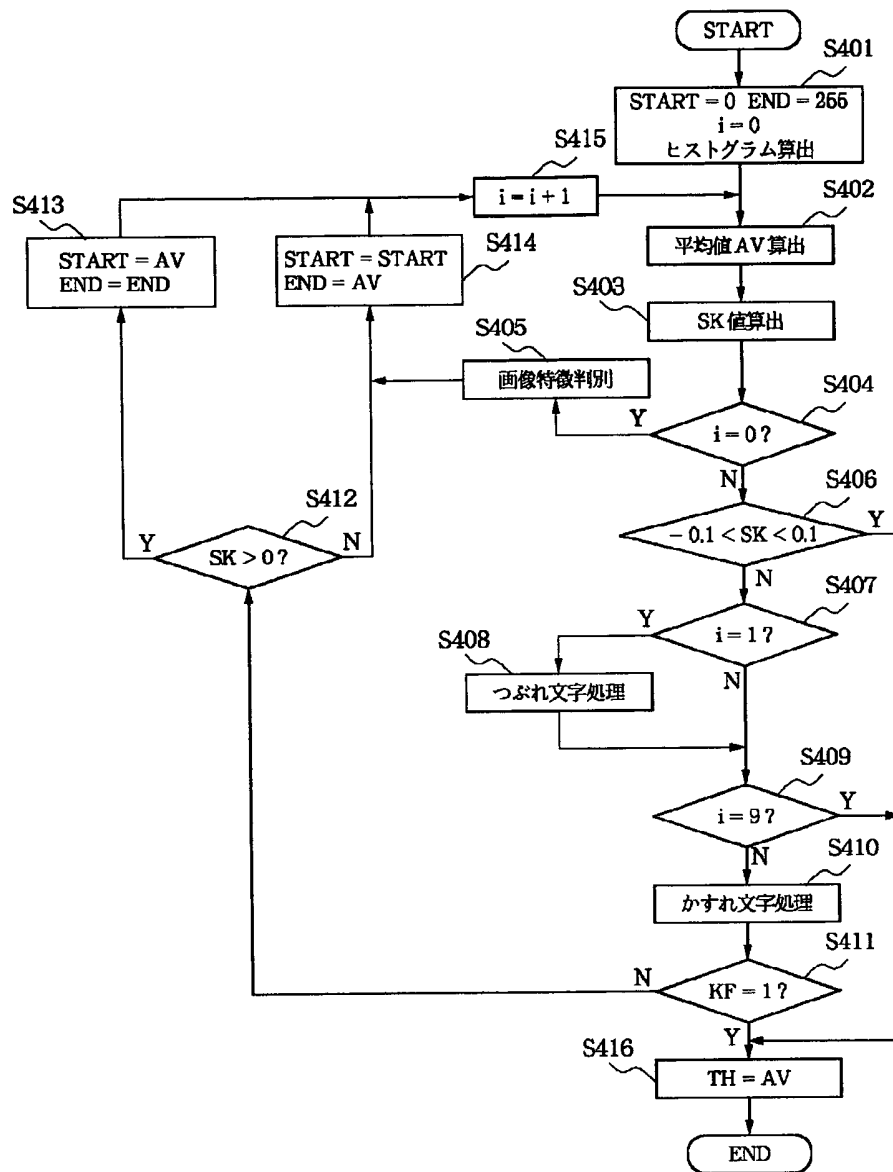
【図10】



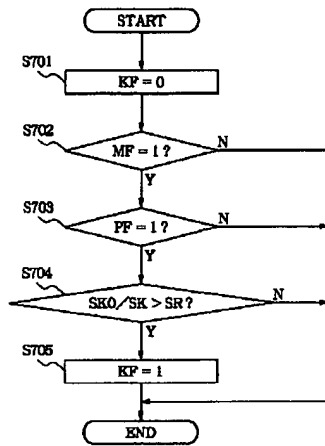
【図11】

	START	END	AV	SK
1回目	0	255	130	-60.2
2回目	0	130	95	-18.3

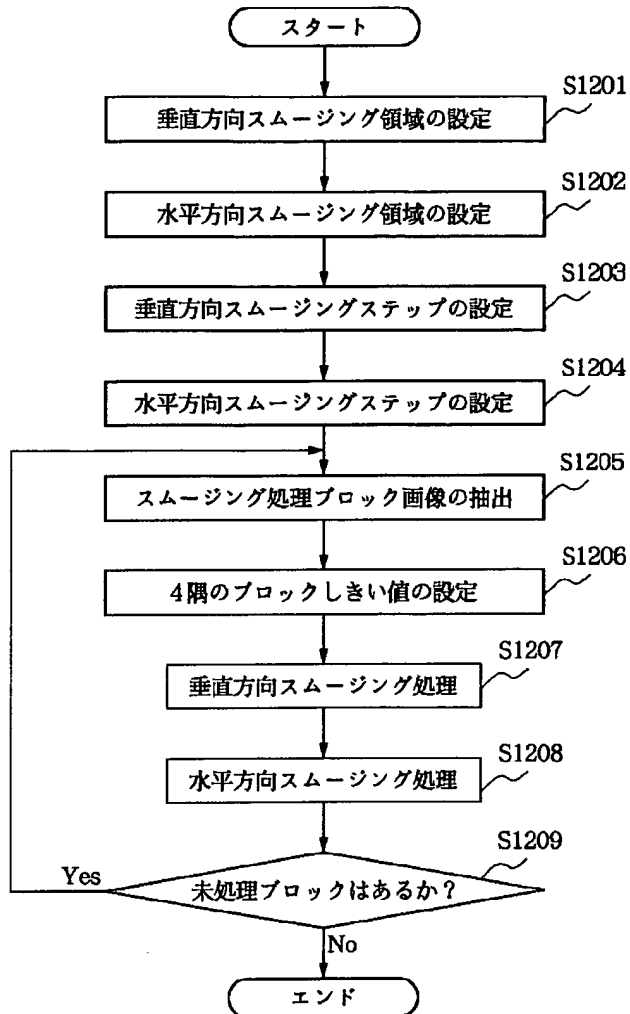
【図4】



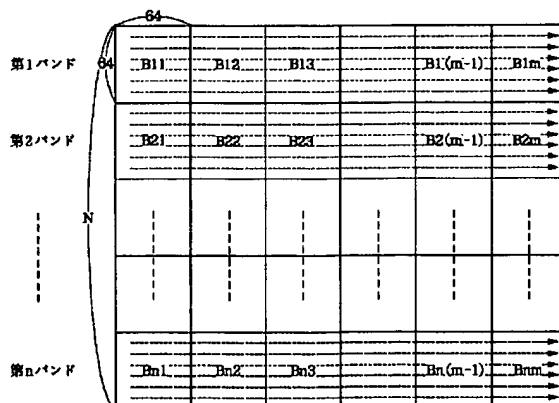
【図7】



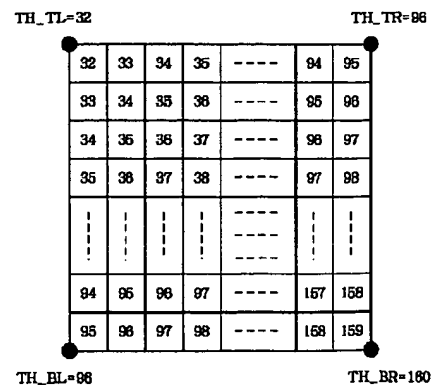
【図12】



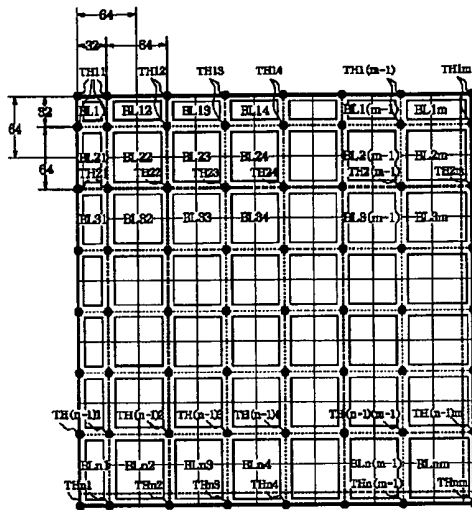
【図13】



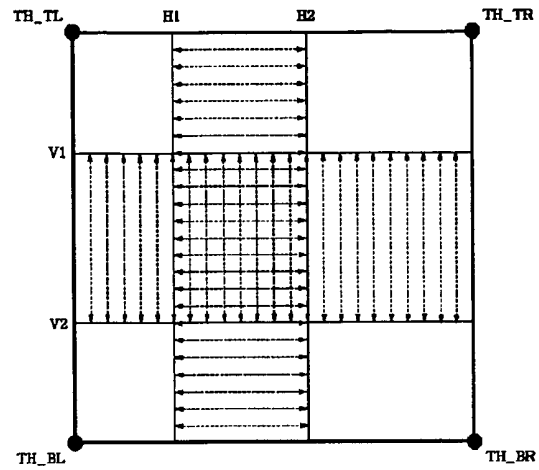
【図16】



【図14】

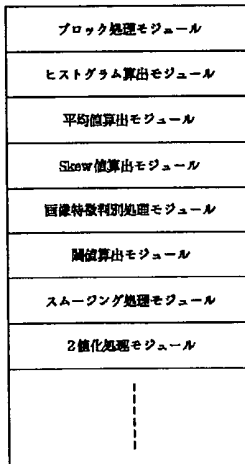
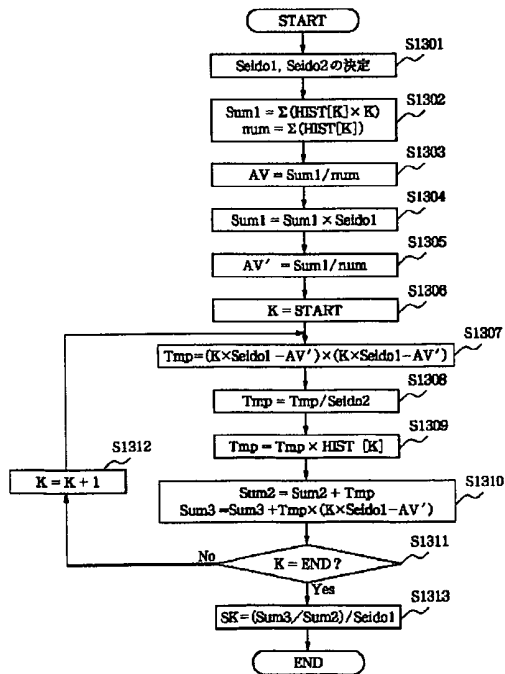


【図15】

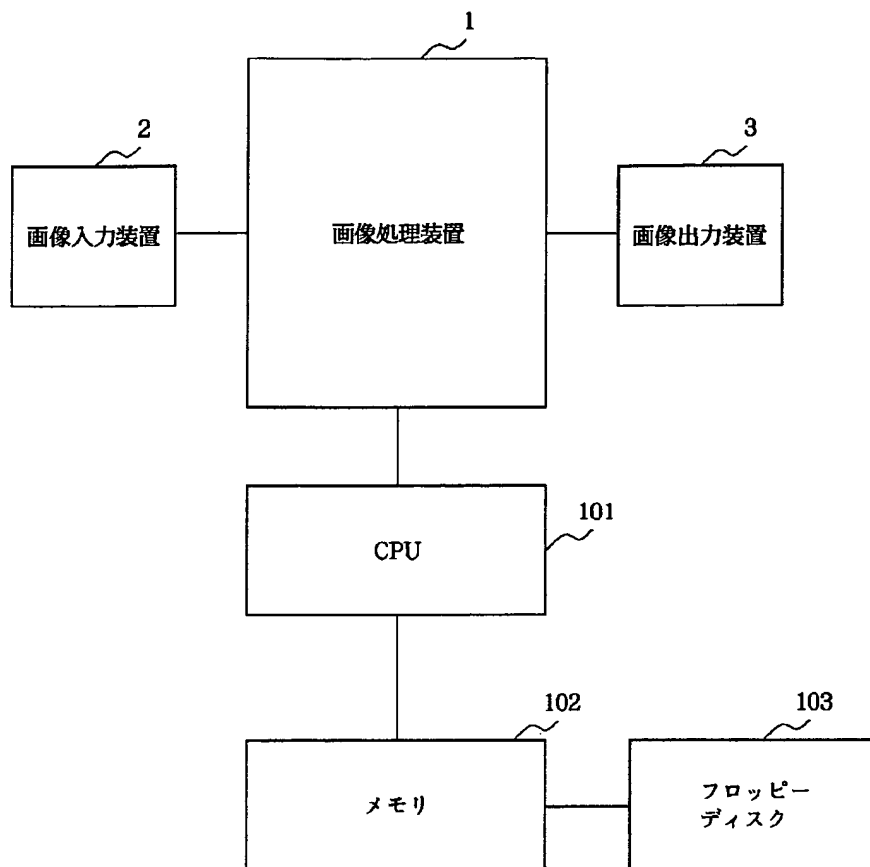


【図19】

【図17】



【図18】



フロントページの続き

(72)発明者 山田 修
東京都大田区下丸子3丁目30番2号キャノ
ン株式会社内

THIS PAGE BLANK (USPTO)